

实验 09_32 位 Timer 的使用

【实验目的】:

通过本实验掌握 LPC1114 的 32 位 Timer 的使用方法。

【实验环境】:

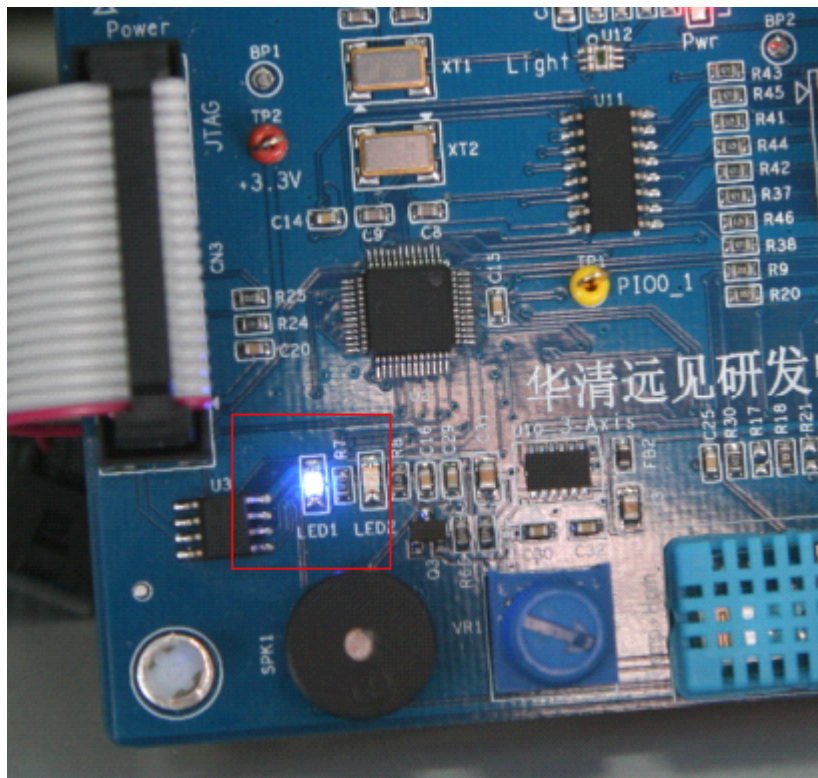
- 1、FS_11C14 开发板
- 2、FS_Colink V2.0
- 3、RealView MDK (Keil uVision4)

【实验步骤】:

- 1、在 Timer32 文件夹下找到并打开 project.uvproj 文件;
- 2、编译此工程;
- 3、通过 FS_Colink 下载编译好的工程到 FS_11C14 开发板;
- 4、按 Reset 键复位开发板, 观察 LED1 现象;
- 5、查看 LPC1114 芯片手册, 学习 32 位 Timer 原理及使用方法。

【实验现象】

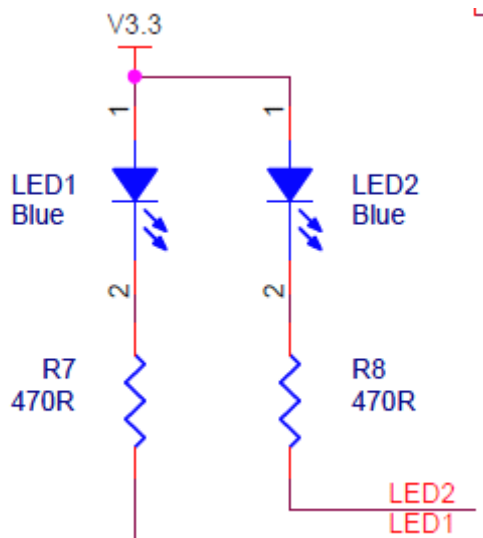
LED1 循环一亮一灭 (如图)



【实验分析】

硬件分析:

在硬件上, 我们将 PIO3_0 引脚连接到 LED1 的控制引脚, 如下图:



由图可以看出当 PIO3_0 输出低电平时，LED1 亮。当 PIO3_0 输出低高电平时，LED1 灭。在程序中我们通过函数 `void GPIOSetValue(uint32_t portNum, uint32_t bitPosi, uint32_t bitVal)` 设置 PIO3_0 的输出值来控制 LED 灯的亮灭。

软件分析：

中断处理函数：

```
void TIMER32_0_IRQHandler(void)
{
    if ( LPC_TMR32B0->IR & 0x01 )
    {
        LPC_TMR32B0->IR = 1;          /* clear interrupt flag */
        timer32_0_counter++;
    }
    if ( LPC_TMR32B0->IR & (0x1<<4) )
    {
        LPC_TMR32B0->IR = 0x1<<4;    /* clear interrupt flag */
        timer32_0_capture++;
    }
    return;
}
```

由中断处理函数可知：当定时/计数器发生一次中断时，`timer32_0_capture++`；会执行一次，再看主程序。

主程序：

```
while(1)
{
    if(timer32_0_counter < 50)
    {
        GPIOSetValue(PORT3, 0, 0);    //点亮 LED1
    }
}
```

```
else if((timer32_0_counter >= 50) && (timer32_0_counter < 100))
{
    GPIOSetValue(PORT3, 0, 1);    //熄灭 LED1
}
else
{
    timer32_0_counter = 0;        //计数器清零
}
}
```

当 timer32_0_capture 的值小于 50 时, LED1 亮, 当 timer32_0_capture 的值大于等于 50 并且小于 100 时 LED1 灭, 否则 timer32_0_capture 清零。